Information Retrieval (CS317)
Programming Assignment No. 2
Spring 2021

Submission Date: April 30, 2021

Assignment Objective

This assignment focuses on Vector Space Model(VSM) for information retrieval. You will be implementing and testing a set of queries using VSM for information retrieval. You need to build a vector space of features using some specified feature selection techniques. The dimension of the space will be R^n, the query is also represented in the same feature space. Cosine similarity is used to compute the similarity between documents and queries. A given threshold can be used to filter the results for a given query. The threshold should be fixed for a given set for queries.

Datasets

You are given a collection of Short Stories (File name: ShortStories) for implementing a tf*idf based vector space. A single file contains a single short story in English. You also need to implement a pre-processing pipeline. It is recommended to first review the given text file for indexing. You need to treat each document as a unique document. This observation offers you many clues for your pipeline implementation and feature extraction. You will be implementing term feature selection based on Term Frequency (tf) and Inverse Document Frequency (idf) scoring. The parameters of tf and idf can be set while creating a specific space of representation. The set of queries are also provided for this assignment. You need to place the queries on the same space and compute the score based on cosine similarity. The weighting scheme used for VSM will be tf*idf, which is a combination of both tf (term frequency of term t in a document) and idf (inverse document frequency computing as (log(df)/ N).

Query Processing

The query processing of VSM is quite tricky, you need of optimize every aspect of computation. The high-dimensional vector product and similarity values of query (q) and documents (d) need to optimized.

Basic Assumption for Vector Space Model (VSM) Retrieval Model

1.Simple model based on linear algebra. Terms are considered as features using a weighting scheme.
2.Allows partial matching of documents with the queries. Hence, able to produce good institutive scoring. Continuous scoring between queries and documents.
3.Ranking of documents are possible using relevance score between document and query.

As we discussed during the lectures, we will implement a VSM Model by selecting features from the document by specifying tf and idf values. You are free to implement a posting list with your choice of data structures; you are only allowed to preprocess the text from the documents in term of tokenization in which you can do case folding, stop-words removal and lemmatization. The stop word list is also provided to you with assignments files. Your query processing routine must address a query parsing, evaluation of the cost, and through executing it to fetch the required list

of documents. The list of documents should be filtered with an alpha value say (α= 0.05), A command line interface is simply required to demonstrate the working model. You are also provided by a set of 10 queries, for evaluating your implementation.

Coding can be done in either Java, C/C++, Python, or C# programming language. There are additional marks for intuitive GUI for demonstrating the working VSM along with free text query search. Coding can be done in either Java, Python, C/C++ or C# programming language. There are additional marks for intuitive GUI for demonstrating the working Vector Space Model along with phrase query search.

Files Provided with this Assignment:

1. ShortStories
2. Stop-words list as a single file
3. Queries Result-set (Gold Standard)

Evaluation/ Grading Criteria

The grading will be done as per the scheme of implementations, query responses and matching with a gold standard (provided query set).

Grading Criteria:
Preprocessing (3 marks)
Formation of Index (1 mark for code complexity 1 mark for saving and loading the indexes)
Vector Space Model (2 marks)
Query processing (2 marks)
Code Clarity (1 mark)
Bonus: GUI (1 mark for making the GUI 1 mark for Good Looking GUI)
The proper clean and well commented code will get 05% more marks.

<The End>